AMENDMENTS TO THE CLAIMS

Claims 1 - 2 (Cancelled).

3. (Currently Amended) In a microprocessor based on superscalar architecture capable of out-of-order execution, comprising: physical registers[[,]] ~~the number of which is greater than that of the logical registers prescribed by the architecture~~; a free list that is designed to hold unallocated physical-register numbers; and a mapping table having entries that are provided in respective correspondence with ~~the~~ a predetermined number of logical registers ~~and that are~~ , said entries being each designed to hold a physical-register number; a method for performing register renaming in a pipelined manner, for each group of instructions that are to go through ~~the~~ a process of register renaming simultaneously, comprising the steps of:

(a) ~~tagging~~ associating each logical-register number shown as a destination operand with a tag based on ~~the~~ order of the instructions in ~~the~~ a group, and ~~tagging~~ associating each logical-register number shown as a source operand that is RAW (read-after-write) dependent on an instruction ~~that goes through the process of register renaming simultaneously~~ of the group with the same tag ~~as for~~ with which the destination operand of said instruction is being associated; and

(b) renaming each ~~tagged~~ logical-register number associated with a tag to the physical-register number that is ~~to be~~ taken out of said free list and is allocated in correspondence with the associated tag, and renaming each ~~non-tagged~~ logical-register number associated with no tag to the physical-register number that is ~~to be~~ obtained by accessing said mapping table.

4. (Currently Amended) In a microprocessor based on superscalar

architecture capable of out-of-order execution, comprising: physical registers [[,]] ~~the number of which is greater than that of the logical registers prescribed by the architecture~~; a free list that is designed to hold unallocated physical-register numbers; and a mapping table having entries that are provided in respective correspondence with ~~the~~ a predetermined number of logical registers ~~and that are~~ , said entries being each designed to hold a physical-register number; a method for performing register renaming in a pipelined manner, for each group of instructions that are to go through ~~the~~ a process of register renaming simultaneously, comprising the steps of:

(a) ~~in the cycle/cycles before the last cycle, tagging~~ associating each logical-register number shown as a destination operand with a tag based on ~~the~~ order of the instructions in ~~the~~ a group, and ~~tagging~~ associating each logical-register number shown as a source operand that is RAW (read-after-write) dependent on an instruction ~~that goes through the process of register renaming simultaneously~~ of the group with the same tag ~~as for~~ with which the destination operand of said instruction is being associated; and

(b) ~~in the last cycle~~ [[,]] renaming each ~~tagged~~ logical-register number associated with a tag to the physical-register number that is ~~to be~~ taken out of said free list and is allocated in correspondence with the associated tag, and renaming each ~~non-tagged~~ logical-register number associated with no tag to the physical-register number that is ~~to be~~ obtained by accessing said mapping table[[.]] ;

wherein said step (a) takes multiple cycles, and said step (b) takes one cycle.